Государственное ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ   
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«Донецкий национальный технический университет»

Д.09.03.04-ПОИС.20-/5847.ЛР

***Кафедра***искусственного интеллекта

и системного анализа

Лабораторная работа №2

по дисциплине: «Основы программирования»

Тема: «Работа с массивами»

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_ асс. А. П. Семёнова

(дата, подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_ асс. И. В. Савицкая

(дата, подпись)

Выполнил:

\_\_\_\_\_\_\_\_\_\_\_\_\_ ст.гр. ПИ-20г М. А. Евсеев

(дата, подпись)

Донецк – 2020

Лабораторная работа №2

Вариант №9

## Задание:

1) Составить метод решения задачи и алгоритм обработки массива, в соответствии с заданием.

2) Написать программу на языке Си.

3) Задание :

Соседями элемента А[i,j] в матрице назовем элементы, расположенные рядом с данным элементом в строке и столбце. Провести операцию сглаживания, которая дает новую матрицу того же размера, каждый элемент которой - среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы.

# Ход работы

## Постановка Задачи

Исходя из матрицы необходимо вывести новый массив того же размера, элементы которого будут являться средним арифметическим имеющихся соседей соответствующего элемента исходной матрицы.

Входные данные: целочисленный массив – arr [N][M].

Промежуточные данные:

* целочисленные константы для обозначения количества строк и столбцов матрицы;
* целочисленное числа для обозначения количества итераций цикла for – i, j;
* целочисленное число для подсчета количества соседей элемента – k.   
   Выходные данные: вещественный массив new\_arr[N][M].

# Метод решения задачи

Среднее арифметическое соседей элемента матрицы – сумма элементов, расположенных рядом с данным элементом в строке и столбце, деленная на количество слагаемых суммы.

Иначе говоря соседи элемента A[m][n] это элементы A[m-1][n], A[m][n-1], A[m+1][n], A[m][n+1], если строка m и столбец n не являются крайними в матрице.

Например, среднее арифметическое соседей элемента матрицы расположенного во 2 строке и 3 столбце A[1][2]=0:

Соседями элемента A[2][3] являются следующие элементы:

7, 4, -2,а их среднее арифметическое - .

Таким образом, чтобы найти среднее арифметическое соседей элемента матрица, мы должны:

1. Убедиться в том, что предполагаемый сосед элемента существует в матрице;
2. Получить сумму существующих соседей;
3. Разделить сумму на количество существующих соседей.

Результат - элемент новой матрицы.

# Алгоритм решения задачи

Введем исходную матрицу размерностью arr[N][M], после чего будем следовать методу решения задачи.

Для того, чтобы определить, не является ли элемент матрицы крайним, достаточно знать размерность матрицы и индексы элемента.

Если индекс равен 0, то есть элемент первый в строке или столбце, то соседа с индексом -1 не существует.

Если разность количества строк или столбцов и индекса строк или столбцов равно 1,то в соседа с индексом превышающего количество строк или столбцов не существует. Например, есть в массиве размерностью A3x7 элемент с индексами A[1][6], расположенный во 2 строке и 7 столбце, будет давать разности (3-1=2) и (7-6=1), то элемента A[1][7] не существует.

В случае, когда мы нашли соседа элемента матрицы, мы складываем соседа с элементом нового массива, который изначально равняется 0 и прибавляем к счетчику соседей единицу k++.

Пройдя 4 проверки на существования элементов и складывая их, мы делим сумму на количество соседей k.

Запишем алгоритм в виде блок-схемы:

arr[i][j]

i=0;i<N;i++

j=0;j<M;j++

N-i=1

k=0

new\_arr[i][j]=0

i=0;i<N;i++

j=0;j<M;j++

M-j=1

i=0

j=0

new\_arr[i][j]+= new\_arr[i-1][j]

k=++

new\_arr[i][j]+= new\_arr[i][j-1]

k=++

new\_arr[i][j]+= new\_arr[i+1][j]

k=++

new\_arr[i][j]+= new\_arr[i][j+1]

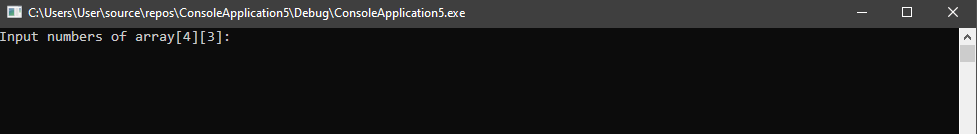
k=++

new\_arr[i][j] = (float) new\_arr[i][j]/k;

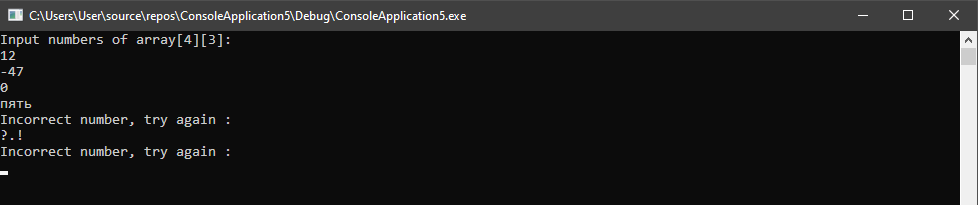
new\_arr[i][j]

# Контрольные примеры

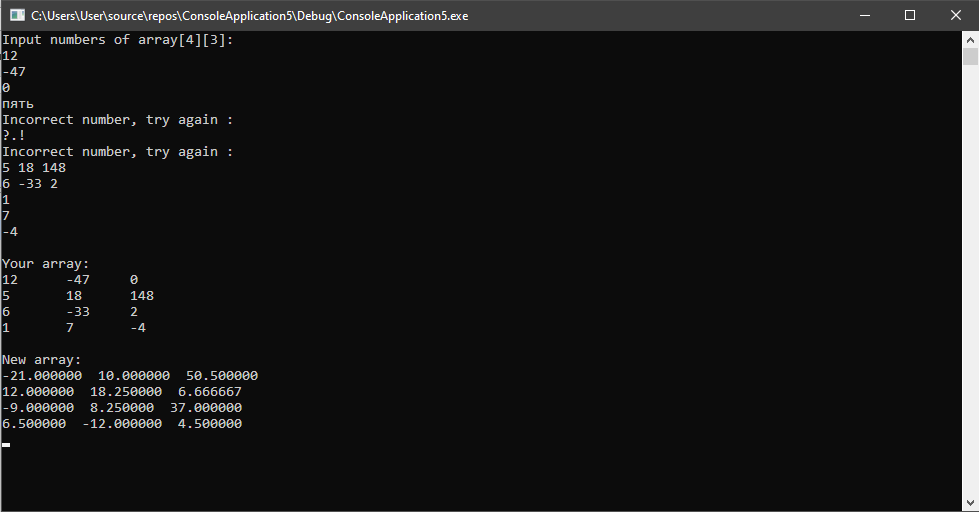
Запустив программу, мы должны ввести элементы матрицы:



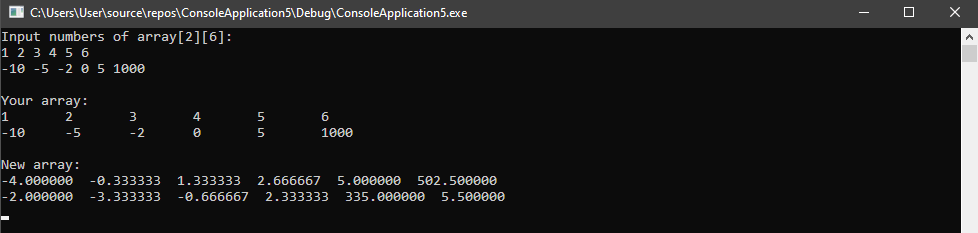
При некорректном вводе программа потребует повторный ввод элемента матрицы:



Введя элементы исходного массива, программа проведет операцию сглаживания и выдаст элементы нового массива, которые равны среднему арифметическому соседей элемента исходной матрицы :



Пример сглаживания матрицы размерностью 2х6:



Чтобы закрыть программу достаточно нажать любую клавишу.

# Листинг программы

Код программы, созданной в Visual Studio 2019:

#include <stdio.h>

#define N 2 //количество строк матриц

#define M 6 //количество столбцов матриц

//Очищаем поток входных данных

**void** flush\_input(**void**)

{

**char** c;

**while** (scanf("%c", &c) == 1 && c != '\n');

}

**int** main()

{

**int** arr[N][M], i, j, k;

//вводим элементы исходного массива

printf("Input numbers of array[%d][%d]:\n", N, M);

**for** (i = 0; i < N; i++)

{

**for** (j = 0; j < M; j++)

{

**while** (scanf("%d", &arr[i][j]) == 0)//обработка в случае неверного ввода числа

{

printf("Incorrect number, try again : \n");

flush\_input();

}

}

}

printf("\nYour array:\n");

**for** (i = 0; i < N; i++)

{

**for** (j = 0; j < M; j++)

{

printf("%d\t", arr[i][j]);

}

printf("\n");

}

//Операция сглаживания матрицы:

**float** new\_arr[N][M]; //инициализация нового массива

printf("\nNew array:\n");

**for** (i = 0; i < N; i++)

{

**for** (j = 0; j < M; j++)

{

k = 0;

new\_arr[i][j] = 0;

//проверки на существование соседей элемента матрицы

**if** (i != 0)

{

k += 1;

new\_arr[i][j] += arr[i - 1][j];

}

**if** (N - i != 1)

{

k += 1;

new\_arr[i][j] += arr[i + 1][j];

}

**if** (j != 0)

{

k += 1;

new\_arr[i][j] += arr[i][j - 1];

}

**if** (M - j != 1)

{

k += 1;

new\_arr[i][j] += arr[i][j + 1];

}

new\_arr[i][j] = (**float**)new\_arr[i][j] / k;

printf("%f ", new\_arr[i][j]);

}

printf("\n");

}

getch();

}

Вывод: Я научился cоставлять методы решения задачи и алгоритмы обработки массива, писать программы на языке Си для решения этих задач, тестировать их и обрабатывать исключения.